

SMUG

BYTES

PRICE
.75

SMUG BYTES
Volume 5, Number 10
October 1988

Subscriptions= \$10.00/year
or free with swap. Send to:

SINCLAIR MILWAUKEE USERS GROUP
P.O. Box 101 Butler, WI 53007

THIS MONTH:

- Bill On Mscript.
- Draw A Halloween Pumpkin
- Rudy's "SQ" NOTES
- Presidents Message
- And Other Great Things

If any articles are copied
please credit SMUG Bytes

NEXT MEETING DATE: 11/02/88

Send all contributions by the
last weekend of the month to:

Bill Heberlein
Editor
SMUG BYTES
5052 N. 91st Street.
Milwaukee, WI 53225

**1988 OFFICERS
& WORKING
MEMBERS**

All Area Codes Are 414

*President - N. Schultz
* - 353 4522
*V. Pres. - W. Heberlein
* - 527 2191
*Secretary - L. Dreger
* - 321 0694
*Treasurer - J. Frohboese
* - 377 1385
*Education - R. Cultice
* - 542 3591

* - Librarians
*Tape/Disk - E. Pawloski
* - 1 338 0260
*Book - D. Nickels
* - 1 284 9691

* - Sub Group Leaders
*Games - N. Schultz
* - 353 4522
Meeting on 3rd Sat. of the month
*Hardware - G. Kraemer
* - 421 0179
*No set date. Call for info.
*Spectrum - R. Hilsmann
* - 251 5291
Meeting on 2nd Wed. of the month
*RP/M - R. Cultice
* - 542 3591
Meeting on 4th Wed. of the month
*QL - R. Hilsmann
* - 251 5291
*Meeting date see Spectrum group *

NOTE NEITHER ANY AUTHOR OR SMUG BYTES TAKES ANY RESPONSIBILITY
FOR ANY HARDWARE MODIFICATIONS TO YOUR EQUIPMENT.)

FROM THE PRES.

Well here is our first meeting with the new date and place. I want to start with a Thank You to Ester for getting us this great location.

Please note the following two ham/computer fest's coming up. First there is the LATE FALL HAM FEST at the Lake County Fairgrounds, routes 45 & 120, Grayslake, IL. 7 am till ? Admission \$3. Date 10/30/88.

The other one is the 6.91 FRIENDLY FEST, Serb Hall, 51st & Oklahoma, Milwaukee. Admission \$3. Date 11/12/88.

Please note Dick has started "C" classes for all those interested in "C" on the QL. The first class was held on Sunday Oct. 2, 1988. For the next date contact Dick Cultice. Also contact Dick if you want any other type of class. He is our Educational Director and he needs the work.

Reminder to all you members dues are coming up for most of you plus the Annual meeting in January.

One more thing please note the change in the Spectrum group meeting date. They will now meet on the second Wednesday of the month instead of the third Wednesday.

GK ENTERPRISES

Your Computer Supply Source

No need to wait for a sale.

Our prices are low everyday

Wide selection of printer ribbons

AXIOM

C. ITOH

EPSON

OKIDATA

PANASONIC

STAR

AND MANY MORE.

SMUG SPECIAL

2700 sheets 9 1/2 x 11 20# Microperf \$23.50

* 546-0111 *

8634 W. National Avenue - West Allis WI 53227

"SQ" NOTES

BY R.A.HILSMANN

Some more ramblings about things not fully covered in the TIMEX MANUAL this month. Looks like the time between News-letters is shrinking, or else I am too busy with other things.

A look at what the Sinclair Manual has to say about Mathematical functions which are also only covered in the Timex Manual in Appendices. Perhaps you feel that this is a waste of time, since most of you have paid attention in High School, and know all about those, but what about the ones of you who did not? Well do not be surprised if you learn something anyways.

Chapter 10 of the Sinclair Manual deals with the mathematics that your computer can handle. It covers the operation \uparrow (raising to the power), the functions EXP and LN, and the trigonometrical functions SIN, COS, TAN and their inverses ASN, ASC, and ATN.

\uparrow and EXP

You can raise one number to the power of another - that means "multiply the first number by itself the second number of times". This is normally shown by writing the second number just above and to the right of the first number; but obviously this would be difficult to do on a computer so the \uparrow symbol was selected instead. For example, the power of 2 are

$$\begin{aligned}2 \uparrow 1 &= 2 \\2 \uparrow 2 &= 2 * 2 = 4 \\2 \uparrow 3 &= 2 * 2 * 2 = 8 \\2 \uparrow 4 &= 2 * 2 * 2 * 2 = 16\end{aligned}$$

At its most elementary level, " $a \uparrow b$ " means "'a' multiplied by itself 'b' times", but this only makes sense if 'b' is a positive number. To find a definition that works for other values of 'b', consider the rule

$$a \uparrow (b+c) = a \uparrow b * a \uparrow c$$

Notice that \uparrow has a higher priority than $*$ and $/$ so that when there are several operations in one expression, the \uparrow s are evaluated before the $*$ s and $/$ s. You should not need much convincing that this works when 'b' and 'c' are both positive whole numbers; but if you decide that you want it to work even when they are not, then you find yourself compelled to accept that

$$a \uparrow 0 = 1$$

$$a \uparrow (-b) = 1/a \uparrow b$$

$a \uparrow (1/b)$ = the bth root of 'a', which is the number that you have to multiply by itself 'b' times to get 'a' and

$$a \uparrow (b*c) = (a \uparrow b) \uparrow c$$

If you have never seen any of this before then don't try to remember it now; just try to remember that

$$a \uparrow (-1) = 1/a$$

and

$$a \uparrow (1/2) = \text{SQR } a$$

maybe when you are familiar with these the rest will begin to make sense.

Experiment with all this by trying this program:

```
10 INPUT a,b,c
20 PRINT a↑(b+c),a↑b*a↑c
30 GO TO 10
```

Of course, if the rule I gave earlier is true, then each time around, the two numbers that the computer prints out will be equal. (Note - because of the way the computer works out \uparrow , the number on the left - 'a' in this case - must never be negative.)

An example of what this function can be used for is that of compound interest. Suppose you keep some of

your money in a savings account which pays 15% interest per year (what a deal). Then after one year you will have not just the 100% that you had anyway, but also the 15% interest your money has earned, making altogether 115% of what you had originally. To put it another way, you have multiplied your sum of money by 1.15. After another year, and if the sum of money you have put in the account originally has not changed, the same will have happened again, so that you will then have $1.15 \times 1.15 = 1.15^2 = 1.3225$ times your original sum of money. In general, after 'y' years, you will have 1.15^y times what you started out with.

If you try this line

```
FOR y=0 TO 100: PRINT y,10*1.15^y:
NEXT y
```

you will see that even starting out with just \$10, it all adds up quite fast, and what's more, it increases faster and faster as time goes on. (Although, you might still find that it doesn't keep up with inflation.)

This type of behaviour, where after a fixed interval of time some quantity multiplies itself by a fixed proportion, is called exponential growth, and it is calculated by raising a fixed number to the power of the time.

Suppose you did this:

```
10 DEF FN a(x)=a^x
```

Here, 'a' is more or less fixed by LET statements: its value will correspond to the interest rate, which changes only ever so often.

There is a certain value for 'a' that makes the function FN 'a' look especially pretty to the trained eye of a mathematician: and this value is called 'e'. Your computer has a function called EXP defined by

```
EXP x = e^x
```

Unfortunately, 'e' itself is not an especially pretty number: it is an infinite non-recurring decimal. You can see its first few decimal places by doing

```
PRINT EXP 1
```

because $\text{EXP } 1 = e^1 = e$. Of course, this is just an approximation. You can never write down 'e' exactly.

LN

The inverse of an exponential function is a logarithmic function: the logarithm (to base 'a') of a number 'x' is the power to which you have to raise 'a' to get the number 'x', and it is written $\log_a x$, normally shown by writing 'a' just below and to the left of 'x'. Thus by definition $a^{\log_a x} = x$; and it is also true that $\log_a (a^x) = x$.

You may already know how to use base 10 logarithms for doing multiplications; these are called common logarithms. Your computer has the function LN which calculates logarithms to the base 'e'; these are called natural logarithms. To calculate logarithms to any other base, you must divide the natural logarithm by the natural logarithm of the base:

$$\log_a x = \text{LN } x / \text{LN } a$$

PI

Given any circle, you can find its perimeter (distance around its edge; called circumference) by multiplying its diameter (width) by a number called PI (Pi is a Greek π , and is used because it stands for the word perimeter).

Like 'e', PI is an infinite non-recurring decimal; it starts off as 3.141592653589... The function PI on your computer returns this number. Try PRINT PI.

SIN, COS & TAN; ASN, ACS & ATN

The trigonometrical functions measure what happens when a point

moves round a circle. Here is a circle of radius 1 (1 what? It does not matter, as long as you keep the same unit all the time. You could also pick any other number!) and a point moving around it. The point started at the 3 o'clock position, and then moves in a counter clockwise direction. I have also drawn in two lines called axes through the center of the circle. The one from 9 o'clock to 3 o'clock is called the x-axis, and the one from 6 o'clock to 12 o'clock is called the y-axis.

To specify where the point is, or how far it has moved around the circle from its 3 o'clock position: let's call this distance 'a'. It is known that the circumference of the circle is 2π (because its radius is 1 and its diameter is thus 2): so when it has moved a quarter way around the circle, $a = \pi/2$; when it has moved half way around, $a = \pi$; and when it has moved all the way, $a = 2\pi$.

Given the curved distance around the circle 'a', two other distances you may like to know are how far the point is to the right of the y-axis, and how far it is above the x-axis. These are called, respectively, the cosine and sine of 'a'. The function COS and SIN on your computer will calculate these.

Note that if the point goes to the left of the y-axis, the cosine becomes negative; and if the point goes below the x-axis, the sine also will become negative.

Another property is that once 'a' has reached 2π , the point is back where it started from and the sine and cosine starts with the same values all over again:

$$\begin{aligned}\text{SIN } (a+2\pi) &= \text{SIN } a \\ \text{COS } (a+2\pi) &= \text{COS } a\end{aligned}$$

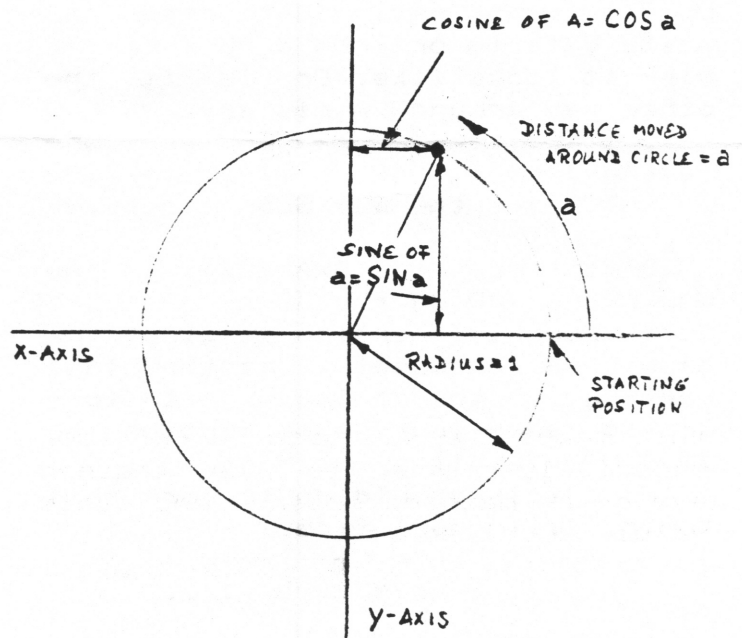
The tangent of 'a' is defined to be the sine divided by the cosine; the corresponding function is called TAN.

Sometimes we have to work these

functions out in reverse, finding the value of 'a' that gives the sine, cosine or tangent. The functions to do this are called arcsine (ASN), arccosine (ACS) and arctangent (ATN).

In the diagram of the point moving around the circle, look at the radius joining the center to the point. You should be able to see that the distance called 'a', the distance that the point has moved around the circle, is a way of measuring the angle through which the radius has moved away from the x-axis. When $a = \pi/2$, the angle is 90 degrees; when $a = \pi$ the angle is 180 degrees; and around to when $a = 2\pi$ ($a = 2\pi$), and the angle is 360 degrees (a full circle). You might as well forget about degrees; and measure the angle in terms of 'a' alone: say then that you are measuring the angle in radians. Thus $\pi/2$ radians = 90 degrees and so on.

Remember that on your computer SIN, COS etc. use radians and not degrees. To convert degrees to radians, divide by 180 and multiply by π ; to convert back from radians to degrees, divide by π and multiply by 180.



Have fun, like always, I hope this has helped to clarify some of the uses for each function.

A SINCLAIR PC?

Just got the new fall catalog from SHARP's. A SINCLAIR MSDOS PC?? You've got to be kidding, \$ 699.95 to \$ 1449.95, depending on what kind of setup you like!

I don't wish to sound negative, but isn't this like inventing the wheel all over again? What happened to being different or unique? Isn't this why we are still stuck on the 2068 or the QL, because they are unique? Unique in that they are more transparent etc.

Why would I buy something just that ordinary, something I don't have to fight to get software for? Something I can buy at the corner computer store? Everybody has them; IBM clones that is, and probably cheaper! I don't see it, but then that's my opinion. Uncle Clyde must be crunching his teeth thinking about such cliché.

But for all you IBM lovers, DISCOVER is available, \$ 49.95 will allow the QL to write or read to any PC disk, and it will also let you view or delete files, but will it let you run PC programs on your QL?? I guess not, just read and write data to or from a PC formatted disk it looks like. Or is it the other way around? A mystery.

But yes, there is another one coming, something called an IBM EMULATOR for the QL! Soon they say.

Don't forget the SWAFFEST at the WAUKESHA EXPO on the 30th of OCT. Just thought I mention it since we hadn't talked about it at the last meeting. Another thing, don't forget to renew your subscription to 2068 UPDATE, it's worth it.

Well this should do it for this month, till' next time,

Your #3 RUDY.....

IMPEX SOFTWARE

P.O. BOX 45

MENOMONEE FALLS, WI. 53051

OLIGER DISK SYSTEM UTILITIES

RECOVER version 1.3

A disk utility which allows you to recover intact data on a corrupted disk formatted with SAFE-DOS. This program will restore a directory to the disk.

\$5.00 INCLUDES INSTRUCTIONS

LIBRARY version 1.5

A disk utility which permits you to catalog all SDOS formatted disks in- to a master Library, can be alphabetically sorted for a printout to either a full size or the 2040 printer, listing will give the name of the program, data type and the name of the disk on which data can be found. Menu driven. Will search for, list and load any or all data by title. Move data function will let you Move data from one disk to another from listings.

\$7.00 INCLUDES INSTRUCTIONS

SG_NOTES PROGRAM'S

All programs previously published in SMUG BYTES - SG NOTES (latest versions), LIBRARY, RECOVER, FILE0 MENU, VFILE+007, VFILE+007S, VCALL and BIORHYTHMS.

\$10.00 INSTRUCTIONS CAN BE FOUND IN SMUG BYTES, OF ADD \$5.00 TO RECEIVE INSTRUCTIONS FOR ALL PROGRAM'S.

ABOVE PROGRAM'S HAVE BEEN RELEASED TO PUBLIC DOMAIN.

OTHER PROGRAMS AVAILABLE

CHECKBOOK & BUDGET MANAGER

This program has been in use since 1982 and is copyrighted. It will keep track of your bank account and your household or other budget like no other. Menu driven, it will keep you informed about the budget status the second you make an entry, lists to the screen or printer, has search functions and will reconcile your account. It categorizes your income and expenses and allocates transactions to up to 26 different accounts, up to 500 entries, instant account balance update, entries will stay visible for four entries, many more features, written in superfast basic, easily modified to your liking. Program is available for the 2068, Spectrum or the ZX81 or 1020 in a slightly different format. Also available for dockbank. If you like to compile this program, please specify which compiler.

\$15.00 TAPE \$ 20.00 EPROM INCLUDES MANUAL.

Please inform when ordering, which computer or disk system it is for, also which version (Dock or Regular etc.) you wish. CHECK OR MONEYORDERS ONLY.

***SEND FOR CATALOG OF OTHER UTILITY PROGRAMS AVAILABLE.**

SEND ALL ORDERS OR INQUIRIES TO THE ADDRESS GIVEN AT THE HEAD OF THIS PAGE.

Bill on Mscript

Well this is a reminder what you see is not necessarily what you want. Watch out. Instructions may call for a special character and most likely the special character is generated by a combination of MSCRIPT keys not gotten from the key board. For example the ↑. If you see this symbol it probably is the SYMBOL SHIFT, CAPS SHIFT, N and not the SYMBOL SHIFT H keys.

I just got my update of MSCRIPT from Jack Duhaney so next month I hope to have some new information for you and a little review of it.

If you use et or ot instead of eb or ob the page numbering will be on the "top of page" not the bottom of the page. Also the \$ is not required if you do not want page numbering. There are other commands that go with these top and bottom commands but to understand them best you should play with them. They are not difficult to use so go at it.

Try the MSCRIPT you'll like it.

Bill

The following are the setups for page numbering.

```
>eb=/      SMUG Bytes/$/      August
1988/
```

sets up a "bottom of page"
line on all even pages.

```
>ob=/      August 1988/$/      SMUG
Bytes/
```

sets up a "bottom of page"
line on all odd pages.



WOULD YOU LIKE TO BE ABLE TO DESIGN SCREEN DISPLAYS
LIKE THIS WITH A MINIMUM OF EFFORT?

WELL NOW YOU CAN!! With just a few keystrokes you too, can design and save screen displays even better than this one! You can even make your own UDCs with the built in UDC design mode. You will have your choice of type styles and fonts to choose from. You can even install a scrolling message with extra large letters! A great attention getter for ads, promotions of shows. You will be amazed at how simple this program is to use! Many built in commands like change ink/paper colors, erase, redo and more! We have never seen anything like it! We'll bet you haven't seen a utility like this either!

ONLY \$14.95+\$2 SH.

AVAILABLE EXCLUSIVELY FROM:
RMC ENTERPRISES
1419 1/2 7TH STREET
OREGON CITY, OR 97045
503/655-7484

The following is a nice little program. It is just in time for the October halloween season.

```

1 REM      PUMPKIN
  © GREG STEINER 1988
10 BORDER 0: PAPER 6: CLS : IN
K 0
20 PLOT 127,160: DRAW 0,-155,1
.2*PI
30 DRAW 0,155,1.2*PI
40 PLOT 127,160: DRAW 0,-155,.
5*PI
50 DRAW 0,155,.5*PI: DRAW 0,-1
55,.9*PI: DRAW 0,155,.9*PI
300 FOR y=175 TO 0 STEP -1
310 FOR x=255 TO 0 STEP -1
320 IF POINT (x,y) THEN GO TO -
20
330 PLOT x,y
340 NEXT x
350 NEXT y
420 FOR y=0 TO 175
430 FOR x=0 TO 255
440 IF POINT (x,y) THEN GO TO 1
80
450 PLOT x,y
470 NEXT x
480 NEXT y
500 REM stem
510 INK 4
520 FOR i=124 TO 135
530 PLOT i,155
540 DRAW 5,19,-PI
550 NEXT i
590 INK 0
600 REM face
610 REM eyes
620 FOR y=100 TO 150
630 FOR x=-40+y TO 210-y
640 PLOT x,y
650 PLOT x+85,y
660 NEXT x
670 NEXT y
700 REM nose
710 FOR y=70 TO 100
720 FOR x=y+45 TO 210-y
730 PLOT x,y
740 NEXT x
750 NEXT y
760 REM mouth
765 PLOT 64,65
770 FOR p=.3 TO .56 STEP .02
780 DRAW 126,0,p*PI
785 DRAW -126,0,-(p+.01)*PI
790 NEXT p
800 LET h$="HH AA PL PL YO
  U E E N "
810 FOR i=4 TO 38 STEP 2
820 INVERSE 1: PRINT AT i/2-1,0
;h$(i-3);AT i/2-1,31;h$(i-2)
830 NEXT i
1000 SAVE "PUMPKIN" LINE 1

```

PUMPKIN CK TYPE report:

1	51	4314
10	2000	2170
20	3711	3711
30	2400	2400
40	3034	3034
50	2000	2000
60	2104	2104
70	2000	2000
80	2000	2000
90	2000	2000
100	2000	2000
110	2000	2000
120	2000	2000
130	2000	2000
140	2000	2000
150	2000	2000
160	2000	2000
170	2000	2000
180	2000	2000
190	2000	2000
200	2000	2000
210	2000	2000
220	2000	2000
230	2000	2000
240	2000	2000
250	2000	2000
260	2000	2000
270	2000	2000
280	2000	2000
290	2000	2000
300	2000	2000
310	2000	2000
320	2000	2000
330	2000	2000
340	2000	2000
350	2000	2000
360	2000	2000
370	2000	2000
380	2000	2000
390	2000	2000
400	2000	2000
410	2000	2000
420	2000	2000
430	2000	2000
440	2000	2000
450	2000	2000
460	2000	2000
470	2000	2000
480	2000	2000
490	2000	2000
500	2000	2000
510	2000	2000
520	2000	2000
530	2000	2000
540	2000	2000
550	2000	2000
560	2000	2000
570	2000	2000
580	2000	2000
590	2000	2000
600	2000	2000
610	2000	2000
620	2000	2000
630	2000	2000
640	2000	2000
650	2000	2000
660	2000	2000
670	2000	2000
680	2000	2000
690	2000	2000
700	2000	2000
710	2000	2000
720	2000	2000
730	2000	2000
740	2000	2000
750	2000	2000
760	2000	2000
770	2000	2000
780	2000	2000
790	2000	2000
800	2000	2000
810	2000	2000
820	2000	2000
830	2000	2000
840	2000	2000
850	2000	2000
860	2000	2000
870	2000	2000
880	2000	2000
890	2000	2000
900	2000	2000
910	2000	2000
920	2000	2000
930	2000	2000
940	2000	2000
950	2000	2000
960	2000	2000
970	2000	2000
980	2000	2000
990	2000	2000
1000	2000	2000

